UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/044,290	01/11/2002	Jagadish Bandhole	020706-000910US	6856

60429 7590 07/13/2006

CSA LLP
4807 SPICEWOOD SPRINGS RD.
BLDG. 4, SUITE 201
AUSTIN, TX 78759

EXAMINER
FOWLKES, ANDRE R

ART UNIT	PAPER NUMBER
2192	

DATE MAILED: 07/13/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 10/044,290	Applicant(s) BANDHOLE ET AL.
	Examiner Andre R. Fowlkes	Art Unit 2192

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *4/21/06*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-21* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-21* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☑ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date 1/30/06

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____.

DETAILED ACTION

1. This action is in response to the amendment filed 4/21/06.

2. Claims 1, 17, 20 and 21 have been amended. Claims 1-21 are pending.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claim 1-21 are rejected under 35 U.S.C. 103(a) as being unpatentable over White, U.S. Patent No. 5,896,530 in view of McNally, et al., (McNally), U.S. Patent No. 6,259,448.

As per claim 1, White discloses a **method of using a dynamic computing environment ("DCE") for a plurality of phases in a software lifecycle,** (col. 1:12-17, "This invention relates ... to a system and method enabling a plurality of computers and associated computer resources, some or all of which may be heterogeneous in configuration (i.e. a DCE), to cooperatively process a variety of (software lifecycle) applications"), **the method comprising:**

- configuring the dynamic computing environment for a first phase in the plurality of phases (col. 3:40-44, "a system and method of computer software architecture for enabling a plurality of computers, and associated computer resources, some or all of which may be heterogeneous in configuration (i.e. a DCE), to cooperatively process applications (i.e. phases)", and col. 7:51-52, "dynamic changes to device configurations"),

-wherein said configuring comprises:

- allocating a first subnet (col. 3:40-44, "a system and method of computer software architecture for enabling (i.e. allocating) a plurality of computers (i.e. subnets), and associated computer resources (i.e. subnets), some or all of which may be heterogeneous in configuration (i.e. a DCE), to cooperatively process applications"),

- allocating a first computing device coupled to the first subnet (col. 3:40-44, "a system and method of computer software architecture for enabling (i.e. allocating) a plurality of computers, and associated computer resources, some or all of which may be heterogeneous in configuration (i.e. a DCE), to cooperatively process applications)"),

- allocating a first storage device coupled to the first computing device (col. 3:40-44, "a system and method of computer software architecture for enabling (i.e. allocating) a plurality of computers, and associated computer resources (i.e. storage), some or all of which may be heterogeneous in configuration (i.e. a DCE), to cooperatively process applications"),

- storing a first set of instructions on the first storage device (col. 3:40-44, "a system and method of computer software architecture for enabling a plurality of computers, and associated computer resources, some or all of which may be heterogeneous in configuration (i.e. a DCE), to cooperatively process applications (i.e. instructions on the storage device)"),

- using the configured dynamic computing environment in the first phase (col. 3:40-44, "a system and method of computer software architecture for enabling a plurality of computers, and associated computer resources, some or all of which may be heterogeneous in configuration, to cooperatively process applications (i.e. phases)"),

- configuring the dynamic computing environment for a second phase in the plurality of phases (col. 3:40-44, "a system and method of computer software architecture for enabling a plurality of computers, and associated computer resources, some or all of which may be heterogeneous in configuration, to cooperatively process applications (i.e. phases)"),

- using the configured dynamic computing environment in the second phase (col. 3:40-44, "a system and method of computer software architecture for enabling a plurality of computers, and associated computer resources, some or all of which may be heterogeneous in configuration, to cooperatively process applications (i.e. phases)").

White doesn't explicitly disclose:

- deallocating one or more of the first subnet, the first computing device, and the first storage device,

-allocating a second subnet subsequent to said deallocating the first subnet,

- allocating a second computing device coupled to the second subnet subsequent to said deallocating the first computing device,

- allocating a second storage device coupled to the second computing device subsequent to said deallocating the first storage device,

- storing a second set of instructions on the second storage device subsequent to said deallocating the first storage device.

However, McNally, in an analogous environment, discloses:

- deallocating one or more of the first subnet, the first computing device, and the first storage device (col. 8:63-9:10, "The configuration (i.e. deallocating one or more subnets) and method begins at step 60 by having an administrator open up a resource modeling desktop (e.g., a deployment task window on the GUI). At step 62, the administrator selects a resource model (i.e. subnet) to be deployed or implements a new model", and McNally disclose allocating and deallocating a number of different subnets at col. 1:42-57, "machines of certain classes share certain problems (i.e. a subnet). To manage such distributed systems, it has been proposed to 'abstract' a given 'resource' in the distributed network into a so-called 'model' (i.e. a subnet) to facilitate administration. Examples of distributed system resources include computer and communications hardware, operating system software, application programs, systems of programs cooperating to provide a service, and the like. Managing resource models (i.e. a number of subnets) provides significant advantages. Thus, for example, by enabling an administrator to characterize the type or class of machine that should receive a particular task, <u>resource model-based management obviates naming a vast host of machines explicitly (in other words resources are grouped in to subnets to aid in efficiently performing tasks)</u> or the distribution of tasks to all machines within a domain," and col. 2:56-63, "The icon representing the resource model is then associated with a selected one of the distributed icons, preferably via a <u>drag-and-drop protocol</u>. When the resource model icon is dropped onto the selected distribution icon, <u>the resource model is deployed in the network (and vice versa)</u>", and the drag-and-drop protocol is a well known and well documented technique for adding and removing icons, which in this case, represent resources in a distributed computing environment),

-allocating a second subnet subsequent to said deallocating the first subnet (McNally discloses multiple subnets at fig. 7:72a-72c, and associated text, e.g. col. 10:55-11:26. Further, McNally disclose allocating a number of different subnets at col. 1:42-57, "<u>machines of certain classes share certain problems (i.e. a subnet).</u> To manage such distributed systems, it has been proposed to 'abstract' a given 'resource' in the distributed network into a so-called 'model' (i.e. a subnet) to facilitate administration. Examples of distributed system resources include computer and communications hardware, operating system software, application programs, systems of programs cooperating to provide a service, and the like. Managing resource models (i.e. a number of subnets) provides significant advantages. Thus, for example, by enabling an administrator to characterize the type or class of machine that should receive a particular task, <u>resource model-based management obviates naming a vast host of machines explicitly (in other words resources are grouped in to subnets to aid in efficiently performing tasks)</u> or the distribution of tasks to all machines within a domain.", and col. 2:56-63, "The icon representing the resource model is then associated with a selected one of the distributed icons, preferably via a <u>drag-and-drop protocol</u>. When the resource model icon is dropped onto the selected distribution icon, <u>the resource model is deployed in the network (and vice versa)</u>", and the drag-and-drop protocol is a well known and well documented technique for adding and removing icons, which in this case, represent resources in a distributed computing environment),

- **allocating a second computing device coupled to the second subnet subsequent to said deallocating the first computing device** (col. 2:56-60:, "The icon representing the resource model is then associated with a selected one of the distributed icons, preferably via a drag-and-drop protocol. When the resource model icon is dropped onto the selected distribution icon, the resource model is deployed in the network", and col. 1:44-51, "To manage such distributed systems, it has been proposed to "abstract" a given "resource" in the distributed network into a so-called "model" to facilitate administration. Examples of distributed system resources include computer and communications hardware, operating system software, application programs, systems of programs cooperating to provide a service, and the like", and col. 2:56-63, "The icon representing the resource model is then associated with a selected one of the distributed icons, preferably via a drag-and-drop protocol. When the resource model icon is dropped onto the selected distribution icon, the resource model is deployed in the network (and vice versa)", and the drag-and-drop protocol is a well known and well documented technique for adding and removing icons, which in this case, represent resources in a distributed computing environment),

- **allocating a second storage device coupled to the second computing device subsequent to said deallocating the first storage device** (col. 2:56-60:, "The icon representing the resource model is then associated with a selected one of the distributed icons, preferably via a drag-and-drop protocol. When the resource model icon is dropped onto the selected distribution icon, the resource model is deployed in the network", and col. 1:44-51, "To manage such distributed systems, it has been proposed to "abstract" a given "resource" in the distributed network into a so-called "model" to facilitate administration. Examples of distributed system resources include computer and communications hardware (i.e. storage devices), operating system software, application programs, systems of programs cooperating to provide a service, and the like", and col. 2:56-63, "The icon representing the resource model is then associated with a selected one of the distributed icons, preferably via a drag-and-drop protocol. When the resource model icon is dropped onto the selected distribution icon, the resource model is deployed in the network (and vice versa)", and the drag-and-drop protocol is a well known and well documented technique for adding and removing icons, which in this case, represent resources in a distributed computing environment),

- storing a second set of instructions on the second storage device subsequent to said deallocating the first storage device (col. 2:56-60:, "The icon representing the resource model is then associated with a selected one of the distributed icons, preferably via a drag-and-drop protocol. When the resource model icon is dropped onto the selected distribution icon, the resource model is deployed in the network", and col. 1:44-51, "To manage such distributed systems, it has been proposed to "abstract" a given "resource" in the distributed network into a so-called "model" to facilitate administration. Examples of distributed system resources include computer and communications hardware, operating system software, application programs, systems of programs cooperating to provide a service, and the like", and col. 2:56-63, "The icon representing the resource model is then associated with a selected one of the distributed icons, preferably via a drag-and-drop protocol. When the resource model icon is dropped onto the selected distribution icon, the resource model is deployed in the network (and vice versa)", and the drag-and-drop protocol is a well known and well documented technique for adding and removing icons, which in this case, represent resources in a distributed computing environment).

Therefore, it would have been obvious to a person of ordinary skill in the art, at the time the invention was made, to incorporate the teachings of McNally into the system of White to have:

- deallocating one or more of the first subnet, the first computing device, and the first storage device,

-allocating a second subnet subsequent to said deallocating the first subnet,

- allocating a second computing device coupled to the second subnet subsequent to said deallocating the first computing device,

- allocating a second storage device coupled to the second computing device subsequent to said deallocating the first storage device,

- storing a second set of instructions on the second storage device subsequent to said deallocating the first storage device.

The modification would have been obvious because one of ordinary skill in the at would have wanted a convenient way of dynamically creating, allocating, using and deleting multiple subnets composed of resources, to fully exploit the advantages of a distributed computing environment, McNally, col. 1:42-62.

As per claim 2, the rejection of claim 1 is incorporated and further, White discloses that **the plurality of phases comprise a development phase** (col. 4:22, "enabling development of application(s)").

As per claim 3, the rejection of claim 2 is incorporated and further, White discloses:

- **using the configured DCE for a first task** (col. 3:40-44, "a system and method of computer software architecture for enabling a plurality of computers, and associated computer resources, some or all of which may be heterogeneous in configuration, to cooperatively process applications (i.e. multiple tasks processed simultaneously)"),

- using the configured DCE simultaneously with the first task for a second task (col. 3:40-44, "a system and method of computer software architecture for enabling a plurality of computers, and associated computer resources, some or all of which may be heterogeneous in configuration, to cooperatively process applications (i.e. multiple tasks processed simultaneously)").

As per claim 4, the rejection of claim 1 is incorporated and further, White discloses that **the plurality of phases comprise an integration phase** (col. 4:25-26, "enabling applications to be tested as large integrated applications").

As per claim 5, the rejection of claim 4 is incorporated and further, White discloses **using the configured DCE for an integration phase comprises:**

- executing the first set of instructions on the first computing device, wherein the first set of instructions causes a first set of information to be transmitted to a third computing device coupled to the first subnet (col. 3:40-44, "a system and method of computer software architecture for enabling a plurality of computers, and associated computer resources, some or all of which may be heterogeneous in configuration (i.e. several subnets), to cooperatively process applications (i.e. execute instructions)"),

- in response to the first set of information, executing a third set of instructions on the third computing device (col. 3:40-44, "a system and method of computer software architecture for enabling a plurality of computers, and associated computer resources, some or all of which may be heterogeneous in configuration (i.e. several subnets), to cooperatively process applications (i.e. multiple tasks processed simultaneously)"),

- monitoring said executing the first and third set of instructions and a result of said executing the third set of instructions (col. 35:57, "a system has ... a transaction processing monitor", and col. 3:40-44, "a system and method of computer software architecture for enabling a plurality of computers, and associated computer resources, some or all of which may be heterogeneous in configuration (i.e. several subnets), to cooperatively process applications (i.e. multiple tasks processed simultaneously)").

As per claim 6, the rejection of claim 1 is incorporated and further, White discloses that **the plurality of phases comprise a testing phase** (col. 4:25-26, "enabling applications to be tested").

As per claim 7, the rejection of claim 6 is incorporated and further, White doesn't explicitly disclose **if said using the configured DCE in the first phase results in an error, re-provisioning a clean environment in the configured DCE during the testing phase.**

However, McNally, in an analogous environment, discloses **if said using the configured DCE in the first phase results in an error, re-provisioning a clean environment in the configured DCE during the testing phase** (col. 10:8-12 "a test is performed to determine whether the target hosts are represented by an existing domain. As used herein, a "domain" represents a set of target nodes for deployment). If the outcome of the test at step 63 is negative, the routine branches to step 64 to create a new domain", and col. 8:63-9:10, "The configuration and method begins at step 60 by having an administrator open up a resource modeling desktop (e.g., a deployment task window on the GUI). At step 62, the administrator selects a resource model to be deployed or implements a new model").

Therefore, it would have been obvious to a person of ordinary skill in the art, at the time the invention was made, to incorporate the teachings of McNally into the system of White to have **if said using the configured DCE in the first phase results in an error, re-provisioning a clean environment in the configured DCE during the testing phase.** The modification would have been obvious because one of ordinary skill in the at would have wanted a convenient way of dynamically creating, allocating, using and deleting multiple subnets and their components, to fully exploit the advantages of a distributed computing environment, McNally, col. 1:42-62.

As per claim 8, the rejection of claim 1 is incorporated and further, White discloses **a beta testing phase, wherein a first user performs said using the configured DCE in the first phase, and a second user performs said using the**

configured DCE in the second phase (col. 4:25-26, "enabling applications to be tested (in multiple phases)", and col. 17:5-10, "Similarly, more than one user may have multiple applications active on multiple systems at any one point in time.", and col. 24:50-60, "modifications of program logic, data base query, panels, and/or any other components of the transaction will always be installed synchronously").

As per claim 9, the rejection of claim 8 is incorporated and further, White discloses that **during the beta testing phase, said configuring the DCE comprises the first user installing the first set of instructions on the DCE and said using the configured DCE comprises the first user beta testing the first set of instructions using the DCE** (col. 18:3-5, "when installing an application in a new system or when redeploying a new release of an application", and col. 4:25-26, "enabling applications to be tested (in multiple phases)", and col. 17:5-10, "Similarly, more than one user may have multiple applications active on multiple systems at any one point in time.", and col. 24:50-60, "modifications of program logic, data base query, panels, and/or any other components of the transaction will always be installed synchronously").

As per claim 10, the rejection of claim 1 is incorporated and further, White discloses **a staging phase** (col. 10:12, "(the system) provides for development of applications that execute under control of the IET through the user interface, and performs background functions at each stage of the application development. These stages can be defined as definition (i.e. staging), composition, construction and deployment").

As per claim 11, the rejection of claim 10 is incorporated and further, White discloses **installing a new version of the first set of instructions and wherein using the configured dynamic computing environment comprises enabling access for at least one user to the new version of the first set of instructions** (col. 18:3-5, "when installing an application in a new system or when redeploying a new release of an application").

As per claim 12, the rejection of claim 1 is incorporated and further, White discloses **a deployment phase** (col. 10:12, "(the system) provides for development of applications that execute under control of the IET through the user interface, and performs background functions at each stage of the application development. These stages can be defined as definition (i.e. staging), composition, construction and deployment").

As per claim 13, the rejection of claim 12 is incorporated and further, White discloses **testing the first set of instructions; and updating the first set of instructions if updates are required** (col. 4:25-26, "enabling applications (i.e. a set of instructions) to be tested", and col. 4:19-20, "providing real time application upgrades").

As per claim 14, the rejection of claim 1 is incorporated and further, White discloses that **the software lifecycle comprises a shrink-wrap lifecycle** (col. 10:12, "(the system) provides for development of applications that execute under control of the IET through the user interface, and performs background functions at each stage of the application development. These stages can be defined as definition (i.e. staging), composition, construction and deployment", and the White system allows all of the operations performed during shrink wrap lifecycle development).

As per claim 15, the rejection of claim 1 is incorporated and further, White discloses that **the software lifecycle comprises a web site lifecycle** (col. 10:12, "(the system) provides for development of applications that execute under control of the IET through the user interface, and performs background functions at each stage of the application development. These stages can be defined as definition (i.e. staging), composition, construction and deployment", and the White system allows all of the operations performed during website lifecycle development).

As per claim 16, the rejection of claim 1 is incorporated and further, White discloses that **the software lifecycle comprises an ASP lifecycle** (col. 10:12, "(the system) provides for development of applications that execute under control of the IET through the user interface, and performs background functions at each stage of the application development. These stages can be defined as definition (i.e. staging), composition, construction and deployment", and the White system allows all of the operations performed during ASP lifecycle development).

As per claims 17-19, this is another method version of the claimed method discussed above, in claims 1-16, wherein all claimed limitations have also been addressed and/or cited as set forth above. For example, see the White/McNally system (e.g. White col. 137:7-144:38 and McNally 1:44-11:26).

As per claim 20, this is an apparatus version of the claimed method discussed above, in claim 1, wherein all claimed limitations, except for the use of virtual computing devices/subnets have also been addressed and/or cited as set forth above. For example, see Whites portable and dynamic distributed applications architecture (col. 137:7-144:38). White discloses **the use of virtual subnets and virtual computing** devices (definition: computing devices and subnets that are part of the DCE but whose resources have not been allocated yet, specification p.7:13-16) at col. 3:40-44, "a system and method of computer software architecture for enabling a plurality of computers, and associated computer resources, <u>some or all of which may be heterogeneous in configuration</u> (i.e. a DCE), to cooperatively process applications (i.e. instructions on the storage device)", and col. 7:51-52, "dynamic changes to device configurations (i.e. allocation of devices)".

As per claim 21, this is a system version of the claimed method discussed above, in claim 1, wherein all claimed limitations have also been addressed and/or cited as set forth above. For example, see the White/McNally system (e.g. White col. 137:7-144:38 and McNally 1:44-11:26).

Response to Arguments

5. Applicant's arguments have been considered but they are not persuasive.

In the remarks, the applicant has argued substantially that:

1) McNally does not disclose the deallocating limitations found in the independent claims, at p. 10:12-22 and 13:2-4.

Examiner's response:

1) The examiner disagrees with applicant's characterization of the applied art. McNally discloses the deallocating limitations at col. 2:56-63, "The icon representing the resource model is then associated with a selected one of the distributed icons, preferably via a <u>drag-and-drop protocol</u>. When the resource model icon is dropped onto the selected distribution icon, <u>the resource model is deployed in the network (and vice versa)</u>", and the drag-and-drop protocol is a well known and well documented technique for adding and removing icons, which in this case, represent resources in a distributed computing environment.

In the remarks, the applicant has argued substantially that:

2) McNally does not disclose allocating a second subnet, at p. 11:11-12, and 13:1.

Examiner's response:

2) The examiner disagrees with applicant's characterization of the applied art. McNally disclose allocating a number of different subnets at col. 1:42-57, "machines of certain classes share certain problems (i.e. a subnet). To manage such distributed systems, it has been proposed to 'abstract' a given 'resource' in the distributed network into a so-called 'model' (i.e. a subnet) to facilitate administration. Examples of distributed system resources include computer and communications hardware, operating system software, application programs, systems of programs cooperating to provide a service, and the like. Managing resource models (i.e. a number of subnets) provides significant advantages. Thus, for example, by enabling an administrator to characterize the type or class of machine that should receive a particular task, resource model-based management obviates naming a vast host of machines explicitly (in other words resources are grouped in to subnets to aid in efficiently performing tasks) or the distribution of tasks to all machines within a domain."

In the remarks, the applicant has argued substantially that:

3) There is no motivation to combine White and McNally references, at p. 12:9-11, 13:1-10 and 14:18-20.

Examiner's response:

3) In response to applicant's argument that there is no suggestion to combine the references, the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988)and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992). In this case the motivation is found at McNally, col. 1:42-57, "Thus, for example, by enabling an administrator to characterize the type or class of machine that should receive a particular task, <u>resource model-based management obviates naming a vast host of machines explicitly</u> or the distribution of tasks to all machines within a domain." <u>In other words resources are grouped in to subnets to aid in efficiently performing tasks.</u>

In the remarks, the applicant has argued substantially that:

4) The examiner has engaged in impermissible hindsight, at p. 12:11-13 and 14:9-10.

Examiner's response:

4) In response to applicant's argument that the examiner's conclusion of obviousness is based upon improper hindsight reasoning, it must be recognized that any judgment on obviousness is in a sense necessarily a reconstruction based upon hindsight reasoning. But so long as it takes into account only knowledge which was

within the level of ordinary skill at the time the claimed invention was made, and does not include knowledge gleaned only from the applicant's disclosure, such a reconstruction is proper. See *In re McLaughlin*, 443 F.2d 1392, 170 USPQ 209 (CCPA 1971).

In the remarks, the applicant has argued substantially that:

5) The White art and McNally art are in different fields of endeavor, at p. 13:20-22.

Examiner's response:

5) In response to applicant's argument that White and McNally are nonanalogous art, it has been held that a prior art reference must either be in the field of applicant's endeavor or, if not, then be reasonably pertinent to the particular problem with which the applicant was concerned, in order to be relied upon as a basis for rejection of the claimed invention. See *In re Oetiker*, 977 F.2d 1443, 24 USPQ2d 1443 (Fed. Cir. 1992). In this case, both patents are directed to distributed computer networking environments, as evidenced by the title, abstract and disclosure of the documents.

Conclusion

6. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Andre R. Fowlkes whose telephone number is (571) 272-3697. The examiner can normally be reached on Monday - Friday, 8:00am-4:30pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571)272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

ARF

[illegible]

TUAN DAM
SUPERVISORY PATENT EXAMINER